論文2003-40SC-3-7

Multiplexer와 AOP를 적용한 GF(2^m)상의 승산기 설계

(The Design of GF(2^m) Multiplier using Multiplexer and AOP)

卞 基 寧 *, 黃 鍾 學 **, 金 興 壽 ***

(Gi-Young Byun, Jong-Hak Hwang, and Heung-Soo Kim)

요 약

본 논문에서는 고속의 연산동작과 낮은 회로 복잡도를 갖는 새로운 GF(2^m)상의 승산기를 제안한다. 유한체 연산은 다항식 승산과 기약다항식을 적용한 모듈러 연산에 의해 전개되며, 본 논문에서는 이 두 과정을 분리하여 다루었다. 다항식 승산연산은 Permestzi의 기법을 토대로 전개하였고 기약다항식은 AOP로 하였다. 멀티플렉서를 사용하여 GF(2^m)상의 승산회로를 구성하였고, 회로 복잡도와 지연시간을 타 논문과 비교하였다. 제안된 승산기는 낮은 회로 복잡도와 지연시간을 보이며, 회로의 구성이 정규성을 가지므로 VLSI 구현에 적합하다.

Abstract

This study focuses on the hardware implementation of fast and low-complexity multiplier over GF(2^m). Finite field multiplication can be realized in two steps: polynomial multiplication and modular reduction using the irreducible polynomial and we will treat both operation, seperately. Polynomial multiplicative operation in this paper is based on the Permestzi's algorithm, and irreducible polynomial is defined AOP. The realization of the proposed GF(2^m) multiplexer-based multiplier scheme is compared to existing multiplier designs in terms of circuit complexity and operation delay time. Proposed multiplier obtained have low circuit complexity and delay time, and the interconnections of the circuit are regular, well-suited for VLSI realization.

Keyword : finite field, multiplexer, all one polynomial, standard basis, multiplier

Ⅰ. 서 론

유한체는 Golois(1811~1832)에 의해 발견된 대수학의 한 분야로 Galois체, 또는 간단히 GF라 하며, 오류정정부호, 스위칭이론 및 암호이론 등의 분야에 널리 적용되고 있는 연산체계이다[1, 2]. 유한체에서 중요하게 다루어지는 연산으로는 가산, 승산, 제산, 승산에 대한 역원 등이 있으며, 회로 복잡도와 처리속도를 고려한 최적의 연산 알고리즘을 찾기 위한 연구가 오랜 기간 지속되고 있다.

* 正會員, 가톨릭大學校 情報通信電子工學部
(School of Information, Communication & Electronics Eng., Catholic Univ.)
** 正會員, 國民體育振興公團 體育科學研究院
(KOREA Sport Science Institute)
*** 正會員, 仁荷大學校 電子工學科
(Dept. of Electronic Eng., InHa Univ.)
接受日字:2003年2月10日, 수정완료일:2003年4月7日

대표적인 유한체 연산 알고리즘 및 구현회로를 간략히 소개하면, 표준기저를 이용한 Laws의 셀 배열 승산기[3]와 Yeh의 시스토릭 승산기[4], 그리고 정규기저를 이용한 Massey-Omura승산기[5]와 이를 VLSI화시킨 Wang[6]의 회로가 대표적이다. 이후 최근까지 다양한 연구결과들이 제안되어 왔으며[7-9], 그 중 Itoh[10]는 모든 항의 계수가 1인 기약다항식(all one polynomial : AOP)을 유한체의 모듈러 연산에 적용하여 회로의 복잡도를 개선할 수 있음을 보였다. 이후, Koc[11]는 정규기저상의 AOP를 적용하여 회로 복잡도를 개선한 승산기를 제안하였고, Lee[12]는 표준기저상의 AOP, ESP조건에서 구현한 비트 병렬형 시스토릭 승산기를 보였다.

한편, 유한체 승산회로와는 별도로 2진수열의 효율적인 승산을 구현하기 위한 컴퓨터 연산 회로의 개발이 진행되었으며[13], 그 중 Pekmestzi[14]는 multiplexer (MUX)와 전가산기를 적용하여 새로운 2진수열 승산 연산회로를 제안하였다. CMOS VLSI 구현에 있어 Pekmestzi의 회로는 회로 복잡도, 빠른 동작속도, 그리고 회로의 정규성에서 VLSI에 적합한 구조를 갖는다.

본 논문에서는 Pekmestzi의 MUX 배열형 2진수열 승산 회로를 유한체 승산연산에 적합하도록 변형하였고, 승산의 결과에 적용하기 위한 모듈러 연산부를 새롭게 추가하여 새로운 유한체 승산 연산회로를 제시하였다. 회로 복잡도를 보다 개선하기 위해 모듈러 연산에 필요한 기약다항식으로 AOP를 적용하였다. 본 논문에서 제안한 GF(2^m) 승산회로는 다항식 승산 연산부와 모듈러 연산부가 독립적으로 구성되며, 빠른 연산동작을 위해 입출력 구조를 병렬로 구성하였다. m에 대하여 일반화된 수식으로 회로의 구성기법을 보였고, 회로 구성에 필요한 소자의 수를 제시하였다. 설계의 예로써 GF(24)상의 승산회로를 보였고, 타 논문과 회로의 구성 및 적용 소자의 수와 지연시간에 대한 비교를 하여 그 결과를 표로 정리하였다. 비교 결과 본 논문에서 제안한 새로운 GF(2^m)상의 승산회로는 CMOS VLSI 설계시 회로 복잡도, 빠른 동작속도, 그리고 회로의 정규성 면에서 VLSI에 적합하다.

본 논문의 구성을 간략히 소개하면 I 장의 서론에 이어, II장에서는 본 논문에서 새롭게 제안한 GF(2^m)상의 승산전개 기법을 보였다. 2장의 논의를 바탕으로 III장에서는 GF(2^m)상의 병렬 승산기를 설계하였다. IV장에서는 본 논문과 타 논문의 승산기들의 구성을 각 항목별로 비교하였으며, 결론으로 본 논문의 끝맺음을 하였다.

II. GF(2^m)상의 승산전개

1. 유한체상의 원소표현과 AOP

유한체[1, 2] GF(2^m)은 양의 정수 m에 대하여 2^m개의 원소들로 구성된 수 체계이며, 그 원소들간의 연산이 사칙연산에 대하여 닫혀있다. GF(2^m)상의 원소들은 원시원소 α에 의해 식 (1)과 같이 나타낼 수 있다.

$$GF(2^m) = 0, \alpha^0, \alpha^1, \cdots, \alpha^{q-2} \mid q = 2^m \qquad (1)$$

한편, 최고차항을 포함한 다항식의 모든 계수들이 1인 다항식을 All One Polynomial, 간략히 AOP라 한다. $m \le 100$인 GF(2^m)상의 기약다항식들 중 AOP가 존재하는 차수 m은 2, 4, 10, 12, 18, 28, 36, 52, 60, 66, 82, 100이다[12]. GF(2^m)상의 임의의 원소 A(α)에 AOP인 F(x)를 적용한 모듈러 연산, 즉 mod F(α)의 결과는 식 (2)와 같다.

$$A(\alpha) = a_{m-1}\alpha^{m-1} + \cdots + a_1\alpha + a_0 \qquad (2)$$

식 (2)의 각 계수들, $a_{m-1}, \cdots, a_0, a_1$은 GF(2)상의 원소들이며, 계수들의 기저들 $\alpha^{m-1}, \cdots, \alpha^2, \alpha, 1$을 표준기저 또는 관용기저라 한다.

정의 1. GF(2^m)상의 임의의 한 원소를 표준기저의 선형결합으로 표현할 때, 이를 최고차 항과 나머지의 이항식으로 표현할 수 있으며, 식 (3)과 같다.

$$\begin{aligned} A(\alpha) &= a_{m-1}\alpha^{m-1} + a_{m-2}\alpha^{m-2} + \cdots + a_1\alpha + a_0 \\ &= a_{m-1}\alpha^{m-1} + A_{m-2}(\alpha) \end{aligned} \qquad (3)$$

$A_{m-2}(\alpha)$의 아래첨자, m-2는 다항식을 구성하는 최고차항의 차수를 표시하며, $A_0(\alpha) = a_0$이다.

GF(2^m)상의 AOP, $F(x) = x^m + \cdots + x^2 + x + 1$에 α를 대입하면 $F(\alpha) = 1 + \alpha + \alpha^2 + \cdots + \alpha^m = 0$이 성립하며, 이를 α^m에 대하여 전개하면 식 (4)와 같다.

$$\alpha^m = \alpha^{m-1} + \cdots + \alpha^2 + \alpha + 1 \qquad (4)$$

식 (4)로부터 m 이상의 차수를 갖는 α는 (m-1) 이하의 α로 표현되며, 1부터 2(m-1)까지의 정수 i를 사용하여 표현하면 식 (5)와 같다.

$$\alpha^{m+i} = \alpha^{i-1} \tag{5}$$

2. 다항식의 승산 전개

표준기저를 적용하여 $(m-1)$차 이하의 다항식으로 표현된 GF(2^m)상의 두 원소 $A(\alpha)$와 $B(\alpha)$를 정의 1을 적용하여 식 (3)과 같이 이항식으로 나타낼 때, 승산 $C(\alpha)=A(\alpha)B(\alpha)$는 식 (6)과 같이 전개된다.

$$\begin{aligned} C(\alpha) &= [a_{m-1}\alpha^{m-1} + A_{m-2}(\alpha)][b_{m-1}\alpha^{m-1} + B_{m-2}(\alpha)] \\ &= a_{m-1}b^{m-1}\alpha^{2(m-1)} \\ &\quad + [a_{m-1}B_{m-2}(\alpha) + b_{m-1}A_{m-2}(\alpha)]\alpha^{m-1} \\ &\quad + A_{m-2}(\alpha)B_{m-2}(\alpha) \end{aligned} \tag{6}$$

식 (6)의 $A_{m-2}(\alpha)B_{m-2}(\alpha)$항을 전개하면 $2(m-2)$차의 다항식이 되며 이를 $C_{2(m-2)}(\alpha)$라 할 때, 식 (7)과 같다.

$$\begin{aligned} C_{2(m-2)}(\alpha) &= [a_{m-2}\alpha^{m-2} + A_{m-3}(\alpha)] \\ &\quad [b_{m-2}\alpha^{m-2} + B_{m-3}(\alpha)] \\ &= a_{m-2}b_{m-2}\alpha^{2(m-2)} \\ &\quad + [a_{m-2}B_{m-3}(\alpha) + b_{m-2}A_{m-3}(\alpha)]\alpha^{m-2} \\ &\quad + A_{m-3}(\alpha)B_{m-3}(\alpha) \end{aligned} \tag{7}$$

동일한 방법으로 식 (7)의 $C_{2(m-3)}(\alpha)=A_{m-3}(\alpha)B_{m-3}(\alpha)$를 전개할 수 있으며, 이러한 재귀연산의 특성을 이용하여 $C(\alpha)$의 일반식을 표현하면 식 (8)과 같다.

$$\begin{aligned} C(\alpha) &= \sum_{j=0}^{m-1} a_j b_j \alpha^{2j} \\ &\quad + \sum_{j=0}^{m-2} [a_{j+1}B_j(\alpha) + b_{j+1}A_j(\alpha)]\alpha^{j+1} \end{aligned} \tag{8}$$

정의 2. 유한체 연산에서 다항식의 가산은 동일한 차수의 계수들을 모듈러 가산하여 이룰 수 있으므로 자리올림은 발생하지 않는다. 따라서, 식 (8)에서 계수가 곱하여진 다항식의 가산을 $Z_j(\alpha)$로 정의하면 식 (9)와 같다.

$$Z_j(\alpha) = a_{j+1}B_j(\alpha) + b_{j+1}A_j(\alpha) \tag{9}$$

정의 2의 $Z_j(\alpha)$로 부터 식 (8)은 식 (10)으로 표현된다.

표 1. $Z_j(\alpha)$의 연산 값

Table 1. The values of $Z_j(\alpha)$.

a_{j+1}	b_{j+1}	$Z_j(\alpha)$
0	0	0
0	1	A_j
1	0	B_j
1	1	$A_j(\alpha) + B_j(\alpha)$

$$C(\alpha) = \sum_{j=0}^{m-1} a_j b_j \alpha^{2j} + \sum_{j=0}^{m-2} [Z_j(\alpha)\alpha^{j+1}] \tag{10}$$

식 (10)의 두 번째 항에서 $Z_j(\alpha)$의 연산은 a_{j+1}과 b_{j+1}의 값에 따라 결정되며, 이를 진리표로 보이면 <표 1>과 같다.

<표 1>에서 a_{j+1}과 b_{j+1}가 모두 1일 때, $Z_j(\alpha)$는 두 다항식의 가산연산, $A_j(\alpha)+B_j(\alpha)$이 되며 이를 $S_j(\alpha)$로 나타내면 식 (11)과 같다.

$$\begin{aligned} S_j(\alpha) &= A_j(\alpha) + B_j(\alpha) \\ &= (a_j\alpha^j + \cdots + a_1\alpha + a_0) \\ &\quad + (b_j\alpha^j + \cdots + b_1\alpha + b_0) \\ &= (a_j \oplus b_j)\alpha^j + \cdots + (a_1 \oplus b_1)\alpha + (a_0 \oplus b_0) \end{aligned} \tag{11}$$

식 (11)에서 사용한 ⊕는 모듈러 가산의 기호이며, 그 연산 결과는 GF(2)의 원소가 된다.

3. AOP를 적용한 모듈러 연산

GF(2^m)상의 승산을 이루기 위해서는 다항식의 승산과 모듈러 연산이 함께 이루어져야 한다. 식 (10)의 $C(\alpha)$는 $2(m-1)$차의 다항식이 되며, 식 (12)와 같다.

$$\begin{aligned} C(\alpha) &= c_{2(m-1)}\alpha^{2(m-1)} + \cdots + c_m\alpha^m \\ &\quad + c_{m-1}\alpha^{m-1} + \cdots + c_1\alpha + c_0 \end{aligned} \tag{12}$$

$C(\alpha)$에 mod $F(\alpha)$를 적용하여 유도한 $(m-1)$차의 다항식을 $P(\alpha)$라 하면 식 (13)와 같다.

$$\begin{aligned} P(\alpha) &= [A(\alpha)B(\alpha)] \bmod F(\alpha) \\ &= [C(\alpha)] \bmod F(\alpha) \\ &= p_0 + p_1\alpha + \cdots + p_{m-1}\alpha^{m-1} \end{aligned} \tag{13}$$

$F(\alpha)$가 AOP일 때, 식 (4)와 (5)를 식 (12)에 적용하면 식 (13)에서 보인 $P(\alpha)$의 각 계수들을 유도할 수 있으며, 그 연산식은 식 (14)와 같다.

$$p_j = c_m \oplus c_j \oplus c_{m+j+1} \; ; \; j \le m-3$$
$$= c_m \oplus c_j \qquad ; \; j = m-2, m-1 \qquad (14)$$

Ⅲ. MUX를 이용한 GF(2^m)상의 병렬 승산회로의 설계

1. GF(2^m)상의 병렬 승산 회로의 구성

본 논문에서는 GF(2^m)상의 승산을 다항식 승산과 모듈러 연산으로 분리하여 다루었고, 그 구현회로 또한 다항식 승산 연산부와 모듈러 연산부로 구분하였다.

다항식 승산 연산부는 식 (10)으로부터 구성되며, 계수들의 곱은 AND 게이트로써 구현할 수 있다. 두 번째 항의 $Z_j(\alpha)$는 a_{j+1}과 b_{j+1}를 데이터 선택단자로 사용한 MUX로 구현할 수 있으며, 그 연산에 앞서 MUX의 입력 중 하나인 $S_j(\alpha)=A_j(\alpha)+B_j(\alpha)$ 연산이 선행되어야 한다. 모듈러 연산부는 식 (14)로부터 구성되며, $C(\alpha)$의 해당 계수들을 XOR를 적용한 모듈러 합에 의해 구현할 수 있다. 본 논문에서 제시한 승산기의 구성도를 <그림 1>에 보였다.

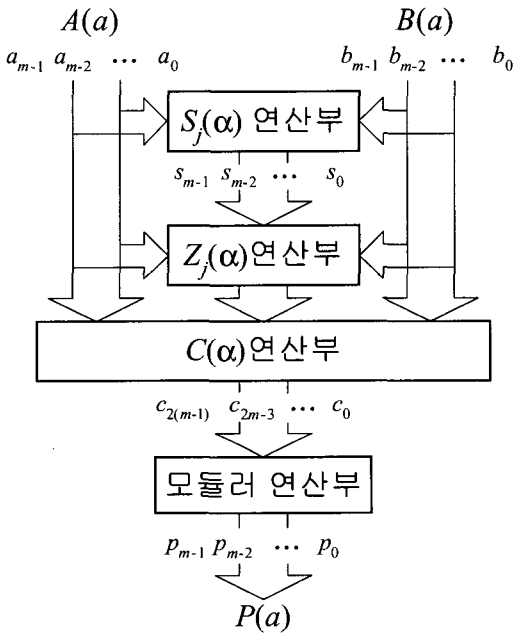


그림 1. MUX를 적용한 GF(2^m)상의 병렬 승산기 구성도

Fig. 1. Block diagram of GF(2^m) parallel multiplier using MUX.

2. 각 연산부별 회로 구성

<표 1>에서 논의한 바와 같이 $a_{j+1}=b_{j+1}=1$일 때의 $Z_j(\alpha)$ 연산의 결과는 $S_j(\alpha)=A_j(\alpha)+B_j(\alpha)$가 되며, GF($2^m$)상의 다항식의 가산은 동일 차수의 계수들 간의 모듈러 합으로 정의된다. 따라서, $S_j(\alpha)$ 연산회로는 $(j+1)$개의 XOR 게이트를 배열하여 구현이 가능하며 <그림 2>와 같다. 본 논문에서는 XOR 게이트를 ⊕로 기호화하였다.

<그림 2>에서 보인 $S_j(\alpha)$ 연산회로의 출력은 이후 $S_{j-1}(\alpha), S_{j-2}(\alpha), \cdots, S_0(\alpha)(=a_0 \oplus b_0)$의 연산에 적용될 수 있다. 따라서, GF($2^m$)상의 승산회로구성을 위한 $S_j(\alpha)$ 연산회로의 구성을 위해 $(m-1)$개의 XOR가 필요하다.

<표 1>과 같이 a_{j+1}와 b_{j+1}를 데이터 선택단자로 활용함으로써 $A_j(\alpha), B_j(\alpha), A_j(\alpha) \oplus B_j(\alpha) = S_j(\alpha)$, 또는 0을 출력하도록 4×1 MUX를 통해 $Z_j(\alpha)$ 연산부를 구성할 수 있다. $A_j(\alpha)$, $B_j(\alpha)$, 그리고 $S_j(\alpha)$는 모두 $(j+1)$개의 항으로 구성된 다항식이므로 그 항의 수만큼 MUX를 배열함으로써 $Z_j(\alpha)$ 연산 회로의 구현이 가능하며 <그림 3>과 같다. <그림 3>에서 s_j와 z_j는 각각 $S_j(\alpha)$와 $Z_j(\alpha)$ 다항식의 각 기저들의 계수를 나타낸다.

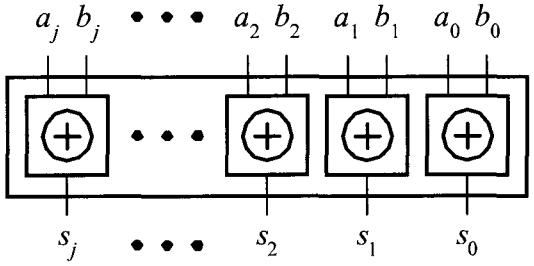


그림 2. GF(2^m)상의 $S_j(\alpha)$ 연산 회로

Fig. 2. Operational circuit of $S_j(\alpha)$ over GF(2^m).

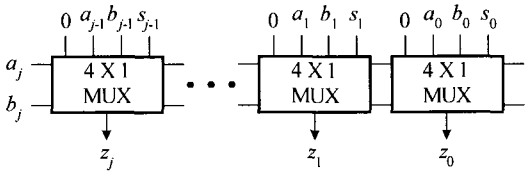


그림 3. GF(2^m)상의 $Z_j(\alpha)$ 연산 회로

Fig. 3. Operational circuit of $Z_j(\alpha)$ over GF(2^m).

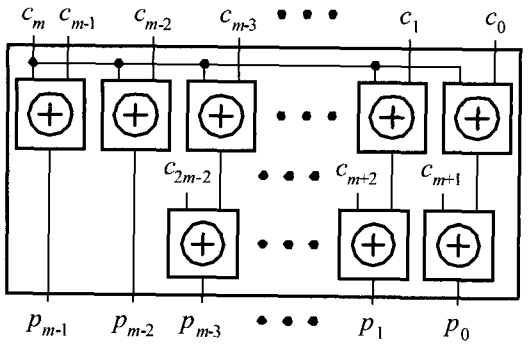


그림 4. MOD 연산 회로
Fig. 4. MOD operational circuit.

<그림 3>에서 보인 4×1 MUX는 모두 a_j와 b_j를 데이터 선택 단자로 사용하며 각 $Z_j(\alpha)$ 연산부에는 j개의 MUX가 사용된다. 따라서, GF(2^m)상의 승산회로구성을 위한 MUX의 수는 $m(m-1)/2$이다.

또한, 각 $Z_j(\alpha)$ 연산의 실행 결과 중복되는 차수항이 발생하며 이러한 중복항들을 모듈러 가산해 주기 위해 $(m-1)(m-2)/2$개의 XOR가 필요하다. $C(\alpha)$에 대한 모듈러 연산의 결과로 유도된 $P(\alpha)$의 각 계수들을 식 (13)에 보였다. 제안된 수식을 바탕으로 모듈러 연산을 수행하는 회로는 <그림 4>와 같다. AOP의 성질에 의해 $j \le m-3$인 조건에서는 삼항의 모듈러 가산이 이루어 지며, $j=m-2$와 $j=m-1$의 조건에서는 이항의 모듈러 가산이 이루어 진다. 따라서, 본 논문에서 제안한 GF(2^m)상의 승산회로를 구성하기 위해 AOP를 적용한 mod $F(\alpha)$연산에서는 $(2m-3)$개의 XOR가 필요하며, 이때 AND는 사용하지 않는다.

3. GF(2^4)에 대한 승산회로의 설계

각 연산부별로 보인 회로블럭들을 결합하여 MUX를 이용한 GF(2^m)상의 병렬 승산회로를 구성할 수 있으며, GF(2^4)를 예로 보였다. 먼저, GF(2^4)상의 두 원소를 각각 $A(\alpha)=a_3\alpha^3+a_2\alpha^2+a_1\alpha+a_0$와 $B(\alpha)=b_3\alpha^3+b_2\alpha^2+b_1\alpha+b_0$라 할 때, 그 승산 $C(\alpha)=A(\alpha)B(\alpha)$는 (14)와 같다.

$$\begin{aligned} C(\alpha) &= \sum_{j=0}^{3} a_j b_j \alpha^{2j} + \sum_{j=0}^{2} [Z_j(\alpha)\alpha^j] \\ &= a_3b_3\alpha^6 + a_2b_2\alpha^4 + a_1b_1\alpha^2 + a_0b_0\alpha^0 \\ &\quad + Z_2(\alpha)\alpha^2 + Z_1(\alpha)\alpha^1 + Z_0(\alpha)\alpha^0 \end{aligned} \quad (14)$$

$Z_2(\alpha), Z_1(\alpha), Z_0(\alpha)$는 각각 $a_3B_2(\alpha)+b_3A_2(\alpha)$, $a_2B_1(\alpha)+b_2A_1(\alpha)$, $a_1B_0(\alpha)+b_1A_0(\alpha)$이 다. $Z_2(\alpha)\alpha^3$는 5, 4, 3의 차수를, $Z_1(\alpha)\alpha^2$는 3과 2의 차수를, 그리고 $Z_0(\alpha)\alpha^0$는 1의 차수를 갖는다. 중복되는 차수의 계수들을 XOR로 모듈러 가산하여 $C(\alpha)$의 각 계수들을 구할 수 있다. 식 (13)으로부터 유도한 $P(\alpha)$의 각 계수들은 $p_3=c_4\oplus c_3$, $p_2=c_4\oplus c_2$, $p_1=c_4\oplus c_1\oplus c_6$, $p_0=c_4\oplus c_0\oplus c_5$이다. 설계된 GF($2^4$)상의 승산회로를 <그림 5>에 보였다.

Ⅳ. 비교 및 검토

GF(2^4)상의 승산회로에 대하여 본 논문과 타 논문에 대한 각 항목별로 비교와 고찰을 하였고, 그 결과를 <표 2>에 정리하였다.

① 구현함수(Function)

본 논문에서는 유한체 상의 두 원소 A, B의 승산에 관하여 논의하였으며 연산함수를 $P=AB$로 하였다. Yeh와 Lee는 승산 및 가산 연산 함수로써 $P=AB+C$에 대한 회로를 제안하였으나, 유한체 연산에서 다항식의 가산은 m개의 XOR를 추가함으로써 간단히 구현되며 주된 연산함수는 승산에 있다.

② 기약다항식(Primitive Polynomial)

GF(2^4)상의 기약다항식 $F(x)$는 x^4+x^1+1, x^4+x^3+1, 그리고 $x^4+x^3+x^2+x^1+1$이 있다. 표준기저를 사용한 승산회로의 경우 $F(x)=x^4+x^1+1$를 주로 적용하고 있으며, 정규기저의 경우 $F(x)=x_4+x_3+1$를 적용한다. Koc와 Lee 그리고 본 논문에서는 AOP로써 $F(x)=x_4+x_3+x_2+x_1+1$를 적용하였다.

③ 입출력형태 (I/O format)

Law와 Yeh는 각각 직렬형과 병렬형 승산기를 함께 제안하였으나, 본 논문과 비교의 일관성을 위해 I/O 형태가 병렬인 회로들에 대해서만 논의하였다.

④ 메모리 소자(Memory)

Yeh는 회로내의 메모리소자를 승산연산에 활용한 시스토릭 승산기를 제안하였고, 하나의 단위 셀에 7개의 메모리소자가 사용되며 GF(2^m)에 적용할 때 $7m^2$개가 사용된다. Wang의 승산회로에서는 2m개의 메모리 소자가 사용된다. Lee의 경우 첫 번째와 두 번째 방법의 회로에서 각각 $4(m+1)^2$개와 $5(m+1)^2$개의 메모리 소자가 사용된다. Koc, 그리고 본 논문에서는 메모리 소자를 사용하지 않는다.

⑤ 가산 게이트(XOR)

본 논문과 비교 논문의 가산 게이트는 모두 2-입력 XOR를 기준으로 하였다. Law와 Yeh의 병렬형 승산회로에서 하나의 단위 셀에 각각 두 개의 XOR게이트가 사용되며 이를 GF(2^m)에 적용할 때 각각 $2m^2$개가 사용된다. Wang과 Koc의 회로에서는 각각 $(2m^2-2m)$개와 (m^2-1)개가 사용된다. Lee의 첫 번째 회로에서는 $(m+1)2$개, 그리고 두 번째 회로에서는 $(m+1)(m+2)$개가 사용된다. 본 논문에서는 $[(m+1)\{(m+2)/2\}-4]$개가 사용된다.

⑥ 승산 게이트(AND)

Law와 Yeh의 병렬형 승산회로에서 하나의 단위 셀에 각각 두 개의 AND게이트가 사용되며 이를 GF(2^m)에 적용할 때 각각 $2m^2$개가 사용된다. Wang과 Koc의 회로에서는 각각 m^2개가 사용된다. Lee의 두 회로는 $(m+1)2$개가 사용되며, 본 논문에서는 m개가 사용된다.

⑦ 데이터 선택기(MUX)

비교 논문에서는 사용하지 않았던 MUX를 본 논문에서는 GF(2^m)상의 승산회로를 구현하기 위해 $m(m-1)/2$개의 4×1 MUX를 사용하였고, 이외의 별도의 제어신호나 메모리소자는 필요하지 않다.

⑧ 지연시간 (latency)

Law, Yeh, Wang, Lee는 단위 연산 셀을 정의한 후, 이들을 배열하여 GF(2^m)상의 승산회로를 구성하였다.

표 2. GF(2^4)상의 승산회로 구성의 비교

Table 2. Comparisons of the related parallel multipliers over GF(2^4)

Multiplier Item	Law[5]	Yeh(2D)[4]	Wang[6]	Koc[11]	Lee[12]		This paper Fig. 5
					Method1	Method2	
1. Function	AB	AB+C	AB	AB	AB+C		AB
2. Polynomial $F(x)$=	x^4+x+1	x^4+x+1	x^4+x^4+1	AOP	AOP		AOP
3. I/O format	parallel	parallel	parallel	parallel	parallel		parallel
4. Memory	-	112	8	-	1 bit latch 100	1 bit latch 125	
5. XOR	32	32	24	15	25	30	11
6. AND	32	32	16	16	25	25	4
7. MUX	-	-	-	-	-	-	6
8. Time delays due to gates	$4(D_A+2D_X)$	$12(D_A+2D_X+2D_L)$	$2(D_A+3D_X)$	D_A+4D_X	$5(D_A+D_X+D_L)$	$5(D_X+D_L)$	D_M+4D_X
Note	D_A = the propagation delay of one 2-input AND gate. D_X = the propagation delay of one 2-input XOR gate. D_L = the propagation delay on one latch. D_M = the propagation delay of one 4×1 MUX.						

따라서, 배열된 연산 셀의 수에 의해 지연시간이 결정된다. Law의 회로는 $m(D_A+2D_X)$, Yeh는 $3m(D_A+2D_X+2D_L)$, Wang은 $(m-2)(D_A+(1+\lceil \log2(m-1)\rceil)DX$, Lee는 두 회로에서는 각각 $(m+1)(D_A+D_X+D_L)$과 $(m+1)(D_X+D_L)$의 지연시간이 발생한다. 다항식의 승산과 모듈러 연산의 회로가 분리되어 구성된 Koc와 본 논문에서는 각각 $D_A+(2+\lceil \log2(m-1)\rceil)D_X$와 $D_M+(2+\lceil \log2(m-1)\rceil)D_X$의 지연시간이 발생한다.

Ⅵ. 결 론

본 논문에서는 MUX와 AOP를 적용하여 GF(2^m)상의 병렬승산을 구현하기 위한 새로운 승산 전개방식과 회로를 제시하였다. 주어진 다항식을 최고차 항과 나머지 항의 이항식으로 정리한 후, 최고차 항의 계수를 나머지 항의 선택단자로 활용하여 승산을 전개하였다. 승산의 전개에서 발생하는 나머지 항의 승산에 동일한 과정을 반복적으로 적용하였다. 또한, 다항식 승산의 결과에 대한 모듈러 연산을 위해 기약다항식을 AOP로 하였다. AOP는 기약다항식의 각 계수들의 값이 설정되어 있으므로 간략화된 회로구성에 적합한 특징을 갖는다.

현재 일반적으로 사용되고 있는 CMOS VLSI 회로 설계 기법[15, 16]에서 4×1 MUX의 경우 8개, 2-입력 XOR와 AND의 경우 각각 3개의 트랜지스터로써 게이트의 구현이 가능하다. <표 2>에서 보인 비교논문과 본 논문의 회로에서 게이트의 수와 CMOS VLSI 구현에 필요한 트랜지스터의 수를 감안할 때, 본 논문에서 제안한 승산회로는 회로 구성 소자의 수에서 상당한 개선 효과가 있다 할 수 있다. 또한, m의 증가에 따라 다항식 승산부와 모듈러 연산부에 필요한 기본 모듈의 확장이 용이하며 회로 소자수의 증가율이 규칙적이므로 VLSI에 유리하다. 소자에 의해 발생하는 지연시간 또한, 비교논문에 비해 우수한 특성을 갖는다.

참 고 문 헌

[1] S.Lin, Error Control Coding, Prentice-Hall, Inc. New Jersey, 1983.

[2] 이만영, BCH부호와 Reed-Solomon부호, 민음사, 1990.

[3] B.A.Laws and C.K.Rushford, "A Cellular-Array Multiplier for GF(2^m)" IEEE Trans. Computer, vol. C-20, no. 12, pp. 1573~1578, Dec. 1971.

[4] C.S.Yeh, I.S.Reed, and T.K.Trung, "Systolic Multipliers for Finite Field GF(2^m)," IEEE Trans. Computer, vol. C-33, pp. 357~360, April 1984.

[5] J.Omura and J.Massey, "Computational Method and Apparatus for Finite Fields," U.S. Patent no. 4,587,627, May 1986.

[6] C.C.Wang, T.K.Trung, H.M.Shao, L.J.Deutsch, J.K. Omura, and I.S.Reed, "VLSI Architecture for Computing Multiplications and Inverses in GF(2^m)," IEEE Trans. Comp., vol.C-34, pp. 709~717, Aug. 1985.

[7] B.Sunar, and C.K.Koc, "Mastrovito Multiplier for All Trinomials," IEEE Trans. Computers, vol. 48, no. 5, pp. 522~527, May 1999.

[8] A.Haibutogullari, and C.K.Koc, "Mastrovito Multiplier for General Irreducible Polynomials," IEEE Trans. Computers, vol. 49, no. 5, pp. 503~518, May 2000.

[9] T.Zhang, and K.K.Parhi, "Systematic Design of Original and Modified Mastrovito Multipliers for General Irreducible Polynomials," IEEE Trans. Computers, vol. 50, no. 7, pp. 734~748, July 2001.

[10] T.Itoh, and S.Tsujii, "Structure of Parallel Multipliers for a Class of Fields GF(2^m)," Information and Computation, vol. 83, pp. 21~40, 1989.

[11] C.K.Koc, and B.Sunar, "Low-Complexity Bit Parallel Canonical and Normal Basis Multipliers for a Class of Finite Fields," IEEE Trans. Computer, vol. 47, no.3, pp. 353~356. March 1998.

[12] C.Y.Lee, E.H.Lu, and J.Y.Lee, "Bit-Parallel Systolic Multipliers for GF(2^m) Fields Defined by All-One and Equally Spaced Polynomials," IEEE Trans. Computers, vol. 50, No.5, pp. 385~393, May 2001.

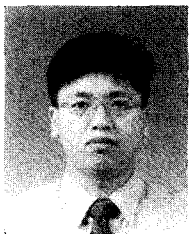
[13] B. Parhami, Computer Arithmetic-Algorithms and Hardware Designs, Oxford University Press, Inc., 2000.

[14] K.Z.Pekmestzi, "Multiplexer-Based Array Multipliers," IEEE Trans. Computer, vol. 48, no.1, pp. 15~23. Jan. 1999.

[15] R.J.Baker, H.W.Li, and D.E.Boyce, CMOS-Circuit Design, Layout, and Simulation, IEEE Press, 1998.

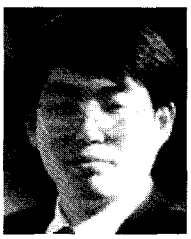
[16] S.M.Kang, and Y.Leblebici, CMOS Digital Integrated Circuits-Analysis and Design, McGraw-Hill, 1999.

저 자 소 개



卞 基 寧(正會員)

1994년 : 인하대학교 전자공학과 공학사. 1998년~2003년 : 공학석사 및 공학박사. 1994년 1월~1996년 8월 : ㈜LG전자 VCR사업부 회로설계연구원. 2003년 3월~현재 : 가톨릭대학교 정보통신전자공학부 강의전담교수. <주관심분야 : 정보이론, 부호이론, 논리 시스템설계, 컴퓨터 구조, 유한체이론의 응용 및 회로구현 등>



黃 鍾 學(正會員)

1988년 : 인하대학교 전자공학과 공학사. 1990년~2001년 : 공학석사 및 공학박사. 1990년~1992년 : ㈜필코 부설연구소 연구원. 1992년~1995년 : ㈜나우정밀 중앙 연구소 전임연구원. 1996년~현재 : 체육과학연구원 책임연구원. <주관심분야 : 이동통신, 스포츠용 기구, VLSI 설계, 모터 자동제어>

金 興 壽(正會員) 第37卷 SC編 第6號 參照
현재 : 인하대학교 전자공학과 교수